DATA REWRITE CONTROL IN DATA TRANSFER AND STORAGE APPARATUS

This invention relates generally to apparatus for transfer and storage of data and, in particular, to apparatus for the transfer and storage of data from a host computing system to a magnetic tape cartridge or the like.

In recent years, it has become increasingly common for companies and other such organisations to back-up their computing systems (and other in-house back-up systems) by storing data on a series of magnetic tape cartridges, for retrieval in the event that the data is lost or corrupted in the primary systems.

An industrial format has been defined for this type of transfer and storage system, in which it is stated that, if a block of data is read back containing one byte in error, it must be rewritten. The industrial format defined for this type of data transfer and storage specifies that as long as all data bits making up a block of data are written correctly at least once, it is not necessary for all data bits to be written correctly on a given rewrite of the block of data. However, in some circumstances, such as the occurrence of a tape defect which spreads later or use of an inferior data reader, using this principle gives a significant possibility that data will not be recoverable from the tape storage means, when required.

We have now devised an arrangement which overcomes the problems outlined above.

Summary of the Invention

In accordance with the present invention, there is provided apparatus for transferring data from a host computing system to one or more magnetic tape storage devices or the like, the apparatus comprising input apparatus for receiving data, dividing it into blocks, and converting said blocks of data to a format suitable for storage on said one or more storage devices, one or more data writers for writing said blocks of data in sets of a plurality of blocks to said one or more storage devices, one or more data readers for reading back data written to said one or more storage devices and transferring said read data to error checking means, said error checking apparatus being arranged to generate a negative output if a block of data includes an error and/or more

than a predetermined number of errors, and control apparatus for causing said one or more data writers to rewrite a set of blocks of data to said one or more storage devices in response to a negative output from said error checking apparatus, said control apparatus being arranged to cause said one or more data writers to rewrite a set of blocks of data to said one or more storage devices until all of the blocks of data in that set are written without error (or with fewer than a predetermined number of errors) during the same rewrite.

Also in accordance with the present invention, there is provided a method of transferring data from a host computing system to one or more magnetic tape storage devices or the like, the method comprising the steps of receiving data and dividing it into blocks, converting said blocks of data to a format suitable for storage on said storage means, writing said blocks of data in sets of a plurality of blocks to said one or more storage devices, reading back data written to said one or more storage devices and transferring said read data to error checking apparatus, said error checking apparatus being arranged to generate a negative output if a block of data includes an error and/or more than a predetermined number of errors, and rewriting a set of blocks of data to said one or more storage devices in response to a negative output from said error checking apparatus until all of the blocks of data in that set are written without error (or with fewer than a predetermined number of errors) during the same rewrite.

The data is preferably written to the tape in codeword quad (or CQ) sets comprising an array, beneficially 2 x 8, of ECC encoded codeword pairs. The apparatus beneficially comprises a history store for storing information relating to at least some of the CQ sets written to the one or more storage devices together with information corresponding to the output of the error checking apparatus for each codeword pair of the set. Each CQ set is preferably identified in the history store by at least 47 bits of data, comprising 1 valid bit, 32 row quality bits (2 row quality bits per codeword pair), 1 dataset bit, 6 cqset bits, 4 acn bits and 3 rotation bits.

The row quality bits for each codeword pair are both beneficially set to 0 when a CQ set is first written to the one or more storage devices. The apparatus preferably includes detection apparatus for determining whether the header of a read codeword pair is correct and, if so, whether each codeword is good or bad, depending on the number of errors they contain, the

apparatus further comprising apparatus for setting the row quality bits to indicate the result of such error checking. In a preferred embodiment, the row quality bits are set to 00 if the header of the codeword pair is corrupted (so that the codeword pair cannot be identified), 01 if the header is correct but both codewords are bad, 10 if the header is correct but only one of the codewords is good, and 11 if the header is correct and both codewords are good.

In a preferred exemplary embodiment of the present invention, the apparatus includes a control register including 1or more (and more preferably 4) bits which can be set by a user to cause a CQ set to always be rewritten unless all of its CQ's are good, regardless of previous rewrites.

Brief Description of the Drawings

An embodiment of the present invention will now be described by way of example only and with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram illustrating the flow of data from a host computing system through a magnetic tape drive according to an exemplary embodiment of the invention;

Figure 2 is a schematic diagram illustrating the structure of a dataset;

Figure 3 is a schematic diagram illustrating the conversion of a dataset to a codeword quad;

Figure 4 is a schematic diagram illustrating the format of data written to a magnetic tape head;

Figure 5 is a schematic diagram illustrating the sequence in which data might be written to a magnetic tape head;

Figure 6 is a table illustrating the operation of apparatus according to an exemplary embodiment of the present invention;

Figure 7 is a schematic block diagram of apparatus according to an exemplary embodiment of the present invention;

Figure 8 is a schematic diagram illustrating a programmable control register for use in apparatus according to an exemplary embodiment of the present invention.

Detailed Description of the Invention

Referring to Figure 1 of the drawings, there is illustrated schematically an exemplary system for transferring data from a host computing system 100 to a magnetic tape cartridge 102. Typically, data is output from the host computing system 100 in short, sharp bursts, whereas it is much more desirable to provide a steady stream of data to a read head for storage on a magnetic tape cartridge 102, in order to minimise wear on the read head motors and optimise the efficiency and storage capacity of the tape cartridge. Thus, data from the host computing system 100 is buffered in a burst buffer 104 before transfer to a logical formatter 106, where data is compressed and converted to a format suitable for storage on the magnetic tape cartridge 102. The logical formatter 106 arranges the data into 'datasets', as described below.

Referring to Figure 2 of the drawings, there is illustrated schematically a dataset 110, consisting of 16 sub datasets 112, each containing 54 rows of data. The data is arranged in the dataset 110 such that the first 468 bytes of a sequence of data are contained in the first row of the first sub dataset $112_{(0)}$, the next 468 bytes of data are contained in the second row of the first sub dataset $112_{(0)}$, and so on until the first sub dataset $112_{(0)}$ is full; then the next 468 bytes of data are contained in the first row of the second sub dataset $112_{(1)}$, etc., so that the last 468 bytes of data in a dataset are contained in the last row of the sixteenth sub dataset $112_{(15)}$.

The datasets 110 are written sequentially into a main buffer 114. As each row of a dataset 110 is written into the main buffer 114, it is notionally split into two sets of data, and 6 parity bytes (Reed-Solomon) are added to each set by a C1 generator to produce two codewords. The bytes of the two codewords in each row are interleaved to produce a matrix of C1 codeword pairs (CCP's) which is stored in the main buffer 114 before transfer to a physical formatter 116.

Datasets 110 are taken sequentially from the main buffer 114 by the physical formatter 116 and written to the magnetic tape 102. Prior to writing the data to the tape 102, the physical formatter 116 adds a 10-byte header 118 to each CCP. It also notionally splits each sub dataset 112 into C2 codewords and adds 10 parity bytes to each. The header 118 consists of, among other things, a dataset number and a CCP designator to indicate which dataset a CCP comes from and where in that dataset the CCP was located. This information is important when it comes to retrieving the data from the magnetic tape. The physical formatter 116 also RLL (run length limited) encodes all data and adds synchronisation fields.

As described above, each row of the matrix is arranged such that it comprises one interleaved codeword pair, the even-numbered bytes forming the first codeword of the codeword pair and the odd-numbered bytes forming the second codeword of the codeword pair.

In a tape drive according to an exemplary embodiment of the invention, there are eight parallel write heads or channels for simultaneously writing data along tape media. Thus, the physical formatter 116 includes a 'CQ writer', which takes each row of a dataset in turn, and converts it into a Codeword Quad (or CQ) set. Referring to Figure 3 of the drawings, a CQ set comprises a 2 x 8 array containing the 16 CCP's in a row of a dataset. Each row of the CQ set is then written to tape via a respective one of the 8 channels. This has the benefit of spreading the C2 codewords along the the full physical length of a dataset on tape, thereby minimising the chance of media defects exceeding the C2 correction budget for any particular codeword. Thus, a dataset is written as 64 CQ sets, as shown in Figure 4 of the drawings, and written to tape, each CQ set being separated by a DataSet Separator (or DSS) tone, the DSS consisting of a repeated binary pattern.

A read head (not shown) follows each write head to read back data just written to the magnetic tape so that the written data can be evaluated for quality. A C1 checker block checks the data read by the read head and determines whether or not there are any errors in the CCP's of each CQ set. If a codeword is found to contain no errors, the C1 checker block returns a positive output for that codeword. If, however, one or more errors are detected in a codeword, the checker block returns a negative output (C1 failure) for that codeword. A write chain controller

(not shown) receives the output from the C1 checker block (known as the CCP RWW status) and, if a CCQ set contains one or more C1 failures, it causes that CCQ set to be rewritten. This process is known as Read While Write (RWW) Retry.

Obviously, there will be some latency, i.e. delay, between the write chain controller initially causing a CQ set to be written to tape, and receiving a negative output from the checker block for that CQ set. Thus, a number of intervening CQ sets will have been written to tape before the faulty CQ set is rewritten. Referring to Figure 5 of the drawings, CQ sets a, b, c and d are sequentially written to tape. By the time CQ set d has been written to tape, the write chain controller has received a number of C1 failures from the C1 checker for CQ set b which exceeds the predetermined threshold, and causes this CQ set to be rewritten. It then resumes sequentially writing CQ sets e and f before receiving sufficient C1 failures (again) from the C1 checker for the rewritten CQ set b, and causing this CQ set to be rewritten once again.

The provision of the above-described RWW function in this type of system necessitates the provision of a history store for storing the history of CQ sets written to the magnetic tape so that it can determine whether a CQ set has been written to the tape and, if so, whether it has been rewritten and how many times, thereby ensuring that all CQ sets have been correctly written to the tape for reliable retrieval of the data when required.

In some systems, the history of CQ sets written to tape is stored in one or more large memory blocks which store the history of all CQ sets written to the tape, in the order in which they were written. When a CQ set is written to tape, it (or at least information identifying it) is stored in the history storage means. As explained above, each CQ set comprises 16 (2 x 8) codeword pairs.

When the information relating to a CQ set is written to the history storage tape, at least one bit is allocated to each codeword pair included therein, such bits being intended to indicate the quality of the codeword pairs as determined during the RWW process described above.

During the RWW process, the error checking block checks each codeword pair for errors and, in the event that an error is detected in a C1 codeword pair, returns a negative output and sets the quality bit in the history storage means to '0'. If no errors are detected in a C1 codeword pair, a positive output is returned by the error checking block and the quality bit in the history storage means is set to '1'. When all of the quality bits for a CQ set are set (or a predetermined number of write cycles have occurred or a predetermined amount of time has elapsed since the CQ set was written to the tape), the system checks the quality bits in the history storage means for that CQ set and, if any errors are indicated (i.e. if any of the quality bits are '0', then the CQ set is rewritten. Particular codeword quads are rotated across tracks (or channels) on each rewrite, to minimise the effect of a particularly bad track. .

An example of such rotation is illustrated by the table in Figure 6 in which a number of CQ sets are rewritten because an error was detected while they were being written. The notation K^x indicates that an error was detected while writing CQ set K. The notation K' indicates that the CQ set K was rewritten once. The notation K" indicates that the CQ set K was rewritten twice, and so on. N is the set number within the data set.

Referring to Figure 7 of the drawings, a write chain controller for use in an exemplary embodiment of the invention comprises an updater 10 which is the entity in charge of deciding which CQ set to write next to a magnetic tape cartridge (not shown) The selected CQ set is sent to a 'next ccp' block 12 for sequencing of CCP's within that CQ set. The 'next ccp' block 12 also requests further CQ sets as necessary.

Information relating to the selected CQ set is also stored in a history array 14, in case it needs to be rewritten later. The history array 14 is essentially a multiport memory, and this array together with the logic closest to it (not shown) form a history entity. Another entity, 'NEW CQ set' 16, provides the updater 10 with the next new CQ set to be written. However, if a previously written CQ set needs to be rewritten, then the updater 10 gives them priority over the new CQ sets.

A quality control block 18 analyses the error checking information returned for each CCP, and produces checkoff signals (to be described later) which are input to the history entity 16. A FWIF block 20 is intended to represent all of the other primary elements of the write chain controller.

As explained above, as soon as a CQ set has been written to the tape, the written data is read back and an error checker is used to check for errors in the CQ set. It checks the header of each CCP in the CQ set against the information stored in the history array. If a match is found, it checks for errors in the codewords of each CCP, and sets the rowqual bits in the history array for each CCP in the matching CQ set according to the result. In the event that the header information read back is corrupted, the rowqual bits for that particular codeword pair are set to 00. If the header information is found to be correct, but both codewords contain errors (or more than a predetermined number of errors), the rowqual bits are set to 01. If the header information is correct and one of the codewords in a codeword pair is good, the rowqual bits are set to 10. Finally, if the header information is correct and both codewords in a codeword pair are good, the rowqual bits are set to 11. This process is known as 'checkoff' and results in a series of 32 bits set to either 1 or 0.

The quality criteria by which a codeword is determined to be good or bad may be rigid, as in some conventional systems, whereby if a C1 codeword is found to have a single error it is marked as being bad. However, in a C1 codeword having 6 parity bytes, it is possible to detect and correct 3 bytes in error during data retrieval, and as such, a C1 codeword having 3 or less errors can be considered "good" because the error correction scheme used in the data retrieval apparatus can handle these.

Whether or not a CQ set needs to be rewritten is determined according to the number of rowqual bits set to '1' for a particular CQ set in the history array after a predetermined number (according to the latency value set for the system) of intervening CQ sets have been written. In the event that a CQ set is rewritten, its information overwrites the row of the history array which contains the information corresponding to the CQ set of which it is a rewrite.

Referring to Figure 8 of the drawings, there is illustrated a control register for use in configuring the write chain controller described above with reference to Figure 7 of the drawings. The 'WCC_CONTROL' register configures the write chain controller function settings. Once writing has commenced, the register should only be written for aborting the write operation. Some of the most relevant bits of the control register will now be described in more detail.

- go: this bit must be set to 1 for the function to be operational. Writing a 0 here aborts the current operation, in which case no further interrupts or communication to C1 will occur and all state machines return to their idle state.
- reserved_bit: this bit is copied into the reserved bit of every codeword pair header.
- strict_rewrites: a 1 here means that a CQ set will only be considered good if all its CQ's are good on the same rewrite. Good CQ's from previous writes of a rewritten CQ set are forgotten (whereas in a conventional system, once a CQ set has been rewritten a maximum of n times, where n = no. of tracks - 1, the CQ set is considered to be 'good' and will not be rewritten again). Setting this bit to 1 may cause extra rewrites because it significantly tightens the quality criteria.
- latency: (also referred to as the 'rewrite distance') this is the number "L" of intervening CQ sets written between two consecutive instances of a rewritten CQ set. The maximum latency allowed by the industrial format referred to above is 6. The minimum latency which can be programmed into this embodiment of the present invention is dependent on the hardware implementation of the tape drive, but is generally of the order of 4 or 5 (although lower values may be programmed if required, particularly for test purposes). Before a new CQ set is written, the write chain controller checks that the CQ set L + 1 CQ sets earlier is safely written to tape, i.e. all of its CQ's are considered to be good according to the programmed quality criteria. If not, the old CQ set is rewritten and the new one is delayed until after the rewrite(s).

In summary, in data transfer and storage operations such as that described above, there is a need to read data back from the tape on which it has been written, in order to determine whether that data needs to be rewritten to the tape because it has not been written to the tape with sufficient

quality to ensure reliable retrieval of the data when required. Information relating to the blocks of data written to the tape is stored together with information output from the error checking block regarding the quality of each codeword pair in a block (or CQ set), according to some predetermined quality criteria. If a CQ in a CQ set is found to be 'bad' it is rewritten over and over again (the tracks on which each CQ is written being rotated for each rewrite) until the whole CQ set has been correctly written to the tape in a single rewrite, thereby substantially improving the quality of the stored data and reducing the probability that data cannot be recovered from the tape, when required. This feature is preferably entirely programmable for use when required.

Although the present invention has been described by way of examples of a preferred embodiment, it will be evident that other adaptations and modifications may be employed without departing from the scope of the invention as defined by the appended claims. Further, the terms and expressions employed herein have been used as terms of description and not of limitation; and, thus, there is no intent to exclude equivalents, but on the contrary it is intended to cover any and all equivalents which may be employed without departing from the scope of the invention as defined by the appended claims.